

CLAIM LISTING**Claim Listing:**

1. (Previously Presented) A method comprising:
assigning each of a plurality of segments comprising a received corpus to a node in a data structure denoting dependencies between nodes;

calculating a transitional probability between each of the nodes in the data structure; and

managing storage of the data structure across a system memory of a computer system and an extended memory of the computer system such that at least one said node is stored in the system memory and another said node is stored in the extended memory simultaneously.

2. (Previously Presented) A method according to claim 1, further comprising:

calculating a frequency of occurrence for each elemental item of the segment; and

removing nodes of the data structure associated with items which do not meet a minimum threshold for the frequency of occurrence.

3. (Original) A method according to claim 2, wherein the frequency of the item is calculated by counting item occurrences throughout the subset and/or corpus.

1 4. (Original) A method according to claim 2, wherein the minimum
2 threshold is three (3).

3
4 Claim 5 (Canceled).

5
6 6. (Previously Presented) A method according to claim 1, wherein the step
7 of managing storage of the data structure comprises:

8 identifying least recently used nodes of the data structure; and
9 storing the least recently used nodes of the data structure in the extended
10 memory of the computer system when the data structure is too large to store
11 completely within the system memory.

12
13 7. (Previously Presented) A method according to claim 6, wherein the
14 extended memory of the computer system comprises one or more files on an
15 accessible mass storage device.

16
17 8. (Original) A method according to claim 7, wherein the data structure
18 represents a language model, spread across one or more elements of a computing
19 system memory subsystem.

20
21 9. (Original) A method according to claim 1, wherein calculating a
22 transition probability includes calculating a Markov transitional probability
23 between nodes.
24
25

1 10. (Original) A storage medium comprising a plurality of executable
2 instructions including at least a subset of which that, when executed by a
3 processor, implement a method according to claim 1.

4
5 11. (Previously presented) A method for predicting a likelihood of an item
6 in a corpus comprised of a plurality of items, the method comprising:

7 building a data structure, across a system memory of a computer system
8 and an extended memory of the computer system, of corpus segments representing
9 a dynamic context of item dependencies within the segments;

10 calculating the likelihood of each item based, at least in part, on a
11 likelihood of preceding items within the dynamic context;

12 iteratively re-segmenting the corpus; and

13 predicting a likelihood of an item in the re-segmented corpus.

14
15 12. (Original) A method according to claim 11, wherein the method of
16 building a dynamic context of preceding dependent items comprises:

17 analyzing the data structure representing the language model; identifying all
18 items with dependencies to or from the item; and

19 using all items with dependencies to or from the item as the dynamic
20 context.

21
22 13. (Original) A method according to claim 11, wherein the language
23 model includes frequency information for each item within the model.

24

25

1 14. (Original) A method according to claim 13, wherein calculating the
2 likelihood of the item comprises:

3 calculating a Markov transition probability for the item based, at least in
4 part, on the frequency of the items comprising the dynamic context.

5
6 15. (Original) A method according to claim 11, wherein calculating the
7 likelihood of the item comprises:

8 calculating a Markov transition probability for the item given the dynamic
9 context of items.

10
11 16. (Original) A storage medium having stored thereon a plurality of
12 executable instructions including instructions which, when executed by a host
13 computer, implement a method according to claim 11.

14
15 17. (Previously Presented) A storage medium comprising executable
16 instructions that are configured to generate, from a corpus, a data structure
17 representing a statistical language model, the data structure for storage across a
18 system memory and an extended memory, the data structure including:

19 one or more root nodes; and

20 a plurality of subordinate nodes, ultimately linked to a root node,
21 cumulatively comprising one or more sub-trees, wherein each node of a sub-tree
22 represents, one or more items of a corpus and includes a measure of a Markov
23 transition probability between the node and another linked node.

24

25

1 18. (Previously Presented) A storage medium according to claim 17,
2 wherein the root node represents a common root item for all subordinate nodes in
3 the one or more sub-trees.

4
5 19. (Previously Presented) A storage medium according to claim 17,
6 wherein the Markov transition probability is a measure of the likelihood of a
7 transition from one node to another node based, at least in part, on the one or more
8 items represented by each of the nodes.

9
10 20. (Previously Presented) A storage medium according to claim 17,
11 wherein the items include one or more of a character, a letter, a number, and
12 combinations thereof.

13
14 21. (Previously Presented) A storage medium according to claim 17,
15 wherein the data structure represents a dynamic order Markov model (DOMM)
16 language model of the textual source.

17
18 22. (Cancelled).

19
20 23. (Previously Presented) A computer system having the storage medium
21 and a processor configured to interpret the computer executable instructions
22 according to claim 17.

23
24 24. (Original) A modeling agent comprising:
25 a controller, to receive a corpus; and

1 a data structure generator, responsive to and selectively invoked by the
2 controller, to assign each of a plurality of segments comprising the received
3 corpus to a node in a data structure denoting dependencies between nodes;

4 wherein the modeling agent calculates a transitional probability between
5 each of the nodes of the data structure to determine a predictive capability of a
6 language model represented by the data structure and iteratively re-segments the
7 received corpus until a threshold predictive capability is reached.

8
9 25. (Previously Presented) A modeling agent according to claim 24, the
10 data structure generator comprising:

11 a dynamic segmentation function, to iteratively re-segment the received
12 corpus.

13
14 26. (Original) A modeling agent according to claim 24, the data structure
15 generator comprising:

16 a frequency analysis function, to analyze a frequency of occurrence of
17 segments within the corpus.

18
19 27. (Previously Presented) A modeling agent according to claim 26,
20 wherein segments that do not meet a frequency of occurrence threshold are
21 removed from the data structure, thus reducing data structure size.

22
23 28. (Previously Presented) A storage medium comprising a plurality of
24 executable instructions including at least a subset of which, when executed,
25 implement a language modeling agent to assign each of a plurality of segments of

1 a received corpus to a node in a data structure denoting dependencies between
2 nodes, and to calculate a transitional probability between each of the nodes in the
3 data structure to determine a predictive capability of a language model denoted by
4 the data structure, wherein the modeling agent dynamically re-segments the
5 received corpus to remove segments which do not meet a minimum frequency
6 threshold.

7
8 Claim 29 (Cancelled).
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25